Benchmark Analysis Report for RTL Simulation Tools

June 25, 2024

- **Benchmark used** is shown in Table 1.
- **Environment setup** is shown in Table 2 and Table 3.
- **Memory access instructions** is shown in Figure 1.
- **Instruction per cycle** is shown in Table 4.

Table 1: Benchmarks

	Category	Benchmark Name	Description	Source
0	SoC	RocketChip	SoC consisting of RocketCore and peripherals.	https://github.com/chipsalliance/rocket-chip
1	SoC	NutShell	RISC-V SoC designed by students in UCAS.	https://github.com/OSCPU/NutShell
2	Processor	RISCVMini	Educational mini RISC-V core.	https://github.com/ucb-bar/riscv-mini
3	Deep learning	Gemmini	Gemmini matrix multiplication core.	https://github.com/ucb-bar/gemmini
4	Floating point	FPU	Floating point unit of RocketChip.	https://github.com/chipsalliance/rocket-chip

Table 2: Platform Settings 1

	CPU	Max Frequency	L1i	L1d	L2	L3
0	12th Gen i7-12700H	4700 MHz	544 KiB (14 instances)	704 KiB (14 instances)	11.5 MiB	24 MiB

Table 3: Platform Settings 2

field	value
OS	Arch Linux (Kernel 6.9.5-zen1-1-zen)
Compiler	gcc 14.1.1 Clang 17.0.6
Verilator	Verilator 5.026
Firtool	Firtool CIRCT unknown git version

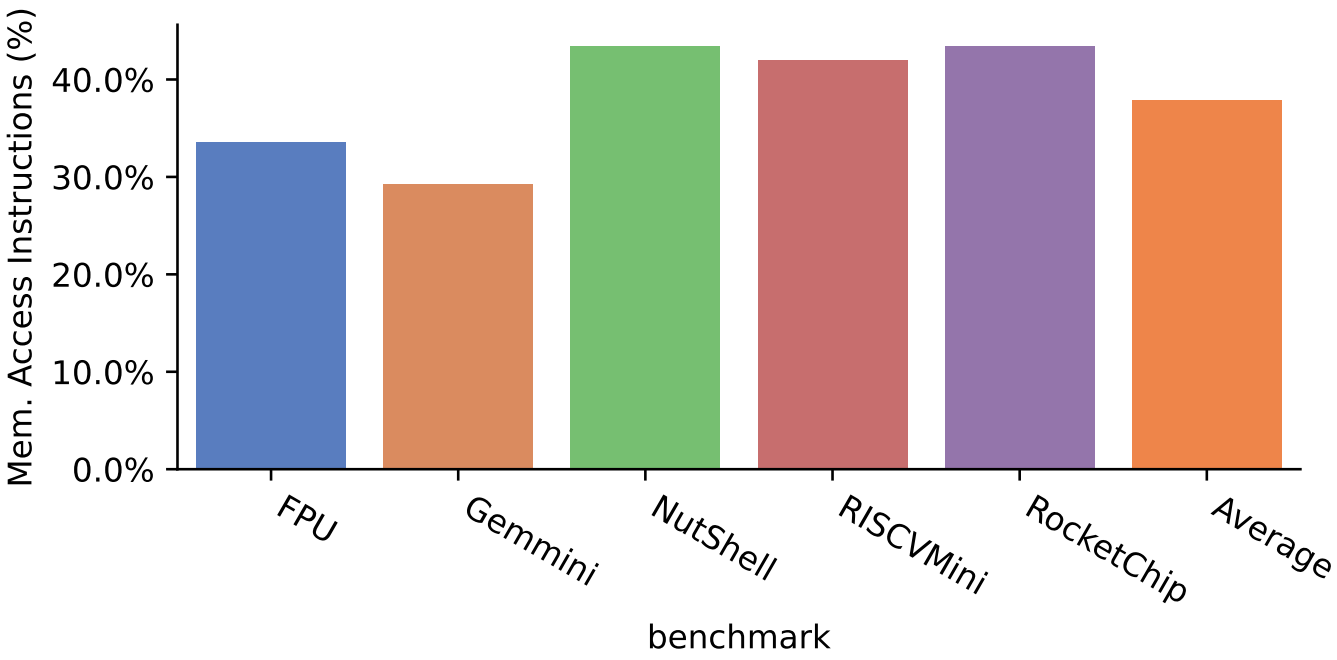

Figure 1: The proportion of data memory access instructions to the total dynamic instructions of Verilator for different benchmarks.

Table 4: Instructions Per Cycle

simulator	**arcilator**	**verilator-1**	**verilator-2**
FPU	3.103181	**3.200152**	2.088054
Gemmini	3.811819	**4.512603**	2.293430
NutShell	**3.194284**	2.390545	1.216389
RISCVMini	3.837050	**3.859147**	1.243605
RocketChip	**3.371983**	2.771164	1.362174

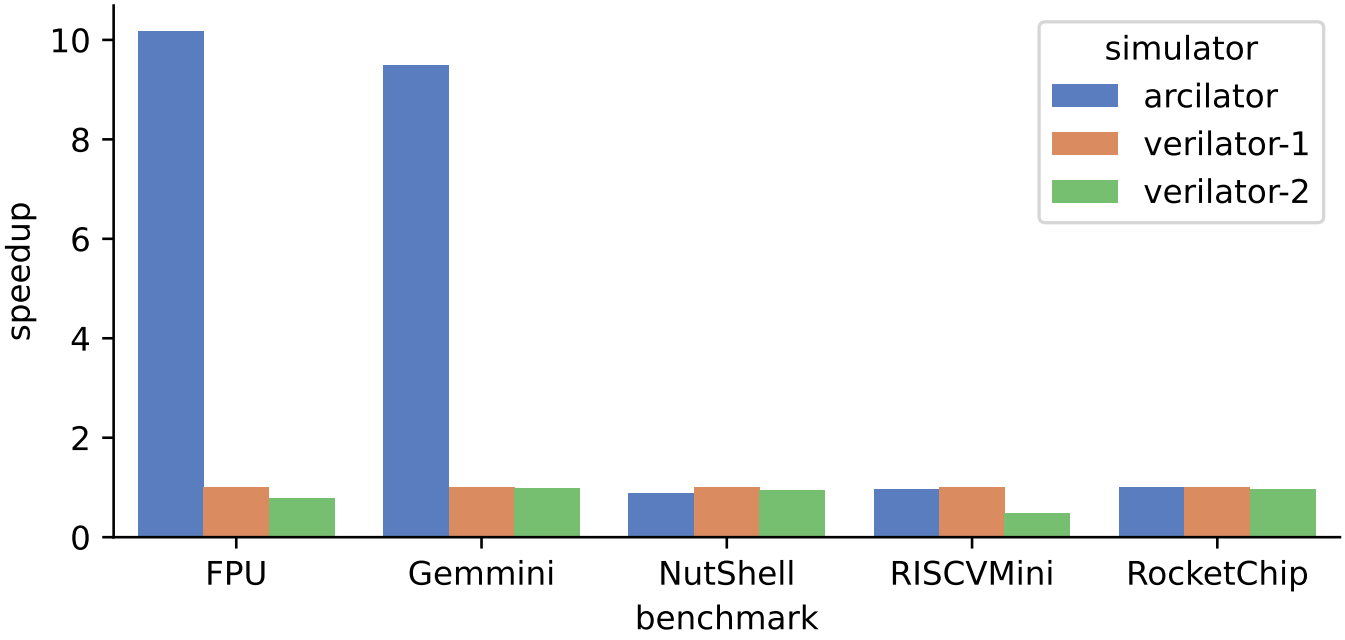

Figure 2: Performance comparison of various simulators relative to the baseline 'verilator-1'.